CENTRE DE RECHERCHE SUR LES TRANSPORTS
(C.R.T.)
CENTRE FOR RESEARCH ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE

CENTRE DE RECHERCHE SUR LES TRANSPORTS
UNIVERSITÉ DE MONTRÉAL
C.P. 6128, SUCCURSALE A.
MONTRÉAL
CANADA H3C 3J7

TÉLÉPHONE: (514) 343-7575
TÉLEX: (21) 05-561325
EMAIL: crt@crt.umontreal.ca
TÉLÉCOPIEUR (FAX): (514) 343-7121

CRT-747

Vehicle Routing Problems Via Tabu Search Metaheuristic

by
V.M. Pureza[1]
and
P.M. França[1, 2]

[1] Faculty of Electrical Engineering, State University of Campinas, C.P. 6101, 13081 Campinas, SP, Brazil.

[2] Invited Researcher at the Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale A, Montréal, Canada H3C 3J7.

Centre de recherche sur les transports — Publication #747

January 1991

Dépôt légal – 1er trimestre 1991

91-02146/

ABSTRACT

This paper addresses the application of the Tabu Search technique to the vehicle routing problem with time windows constraints. A simple adaptation of Tabu Search which uses only a short term memory strategy to overcome local optimality is developed. At first, an initial solution is obtained by a sequential route construction algorithm. After that, an arc interchange improving procedure is applied using as move attributes the deleted edges and the added edges. A sensitivity analysis is carried out in order to establish good choices for the number of edges considered tabu in a movement and for the lengths of the tabu lists. The method is tested on a large set of randomly generated routing problems and on a set of classical test problems.

Key words: Vehicle Routing Problem; Tabu Search; Approximate Methods.

RÉSUMÉ

Cet article porte sur l'application de la méthode de recherche avec tabous ("Tabu Search") au problème de confection de tournées de véhicules. Nous développons une adaptation simple de la recherche avec tabous dans laquelle seule une stratégie de mémoire à court-terme est utilisée pour dépasser les optima locaux. Au départ, une solution initiale est produite par un algorithme séquentiel de construction de routes. Par la suite, on applique une procédure d'amélioration par échange d'arcs dans laquelle les attributs des mouvements sont les arcs ajoutés et éliminés. Une analyse de sensibilité a été effectuée pour déterminer de bonnes valeurs pour le nombre d'arcs qui doivent être considérés tabous dans un mouvement, et pour la longueur des listes de tabous.

La méthode a été testée sur un grand jeu de problèmes pseudo-aléatoires et sur un ensemble de problèmes-tests classiques.

Mots-clés: problèmes de confection de routes; recherche avec tabous; méthodes approchées.

VEHICLE ROUTING PROBLEMS VIA TABU SEARCH METAHEURISTIC

V. M. Pureza *P. M. França*

Faculty of Electrical Engineering
State University of Campinas
C. P. 6101 - 13081 - Campinas, SP
Brazil

1 INTRODUCTION

The Vehicle Routing Problem is basically concerned with defining efficient routes for a fleet of vehicles which have to deliver quantities of goods from a depot to a number of customers. It is assumed that all vehicle tours start and terminate at the depot. Each vehicle has a known capacity and the deterministic load to be delivered to each customer is supplied by exactly one vehicle. The objective is to minimize the total distance traveled by the vehicles. It is also assumed that each vehicle may not exceed a maximum route time (or length). Some customers may impose time windows constraints, i. e., the delivery must occur between an earliest time and a deadline. Real life applications of this problem include deliveries of products to stores, newspaper or mail distribution, preventive maintenance inspection tours, the pickup of children by school buses, etc. Various methods have been proposed for such task, but given the computational effort required, larger routing problems (50 customers or more) are solved by non optimal procedures [1].

Among these non optimal procedures, we shall focus on route improving methods. These methods can be seen as a specific application of a class of heuristic procedures, used for the resolution of many combinatorial optimization problems - the hill climbing heuristics.

The hill climbing heuristics are characterized by the generation of an initial feasible solution, followed by the application of search mechanisms that alter the initial solution. These mechanisms promote essentially the improvement of the objective function towards a local minimum. At this stage, given the lack of improving movements, the algorithm stops.

Despite the excellent performance of these heuristics, there is a major limitation concerning combinatorial problems. Given their non convexity, the local optimum achieved may not be the global optimum. It also implies that following this approach, the quality of the final solution depends critically on the starting solution.

Many techniques have been devised to deal with the limitations of local optimality. Most of them use randomization recourses (e.g. random restart and random shake-up) as a way to find a different path or to delay reaching an optimal point [7].

A different approach for dealing with such limitation is by the application of the *Tabu Search* strategy [7,8,9]. Rather than avoiding local minima, the tabu search, in fact, overcomes it, making possible the continuity of the explorations beyond the local optimum.

2 THE TABU SEARCH TECHNIQUE

The first step for continuing the search is naturally the emergence from the local optimum where the search is being trapped. Given the new conditions, the greedy orientation of hill climbing heuristics is adapted in a way that the least disimproving movement is performed. At this point if we evaluate the available movements and perform the one that provides the greatest improvement in the objective function, it is possible that this movement be the reversal of the movement just done, which implies the return to the local minimum. The next evaluation would again indicate the first movement and, as a result, the search process would consist of a sequence of one movement and its reversal. This situation suggests the first element a strategy must have if it is to overcome local minima: it is necessary to restrict the movements that result in solutions previously found.

On the other hand, if we consider the search process from an initial solution as a sum of alterations in its configuration, it is assumed that the possibility of returning to the initial solution decreases with successive alterations. This premise suggests that the restriction applied previously to certain movements can be relaxed after a certain number of iterations. Rather than a possibility, such relaxation is desirable as those movements do not represent risk of cycling and provide

a greater scope of choices. This is the second element: the liberation of restricted elements that do not risk the continuity of the search.

The tabu search joins these two elements in a very simple and elegant way - the *tabu list*. In this list (or lists) the reversal of movements already done are recorded. These movements are called tabu and they are prohibited to be performed during a number *t* of iterations from the time of the recording. After these *t* iterations, they receive the status of free movements and can be performed again if the evaluation function elects them.

3 AN APPLICATION OF TABU SEARCH TO VEHICLE ROUTING PROBLEMS

3.1 ORIGINAL ALGORITHM'S DESCRIPTION

The algorithm used consists originally of two basic steps:

1. Generation of an initial solution
2. Route improving procedures from the initial solution to a local optimum

In step 1, we used a sequential route construction algorithm [10]. This heuristic was designed for routing problems with time window constraints and it is embodied with mechanisms for time feasibility checking.

In step 2, we applied a node interchange procedure based on M. Dror and L. Levy's algorithm for inventory routing [4]. Basically we followed the authors' ideas of a route improving procedure consisting of two types of inter-route operations - the insertion of a single node and the exchange of two nodes.

The figures below show examples of these two types of operations:

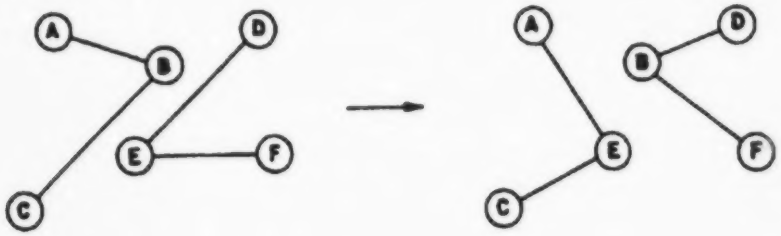

Figure 1 - Exchange of two nodes

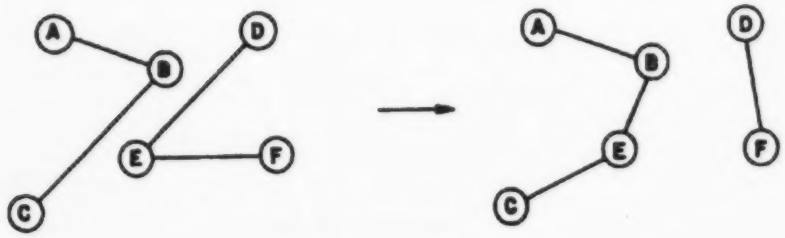

Figure 2 - Insertion of one node

At each iteration, it is computed all feasible operations of these two types and performed the one which gives the greatest improvement in the objective function value, i. e., the total mileage computed for all vehicle routes.

3.2 MODIFIED ALGORITHM'S DESCRIPTION

The modified version of the algorithm presented in section 3.1 as a result of the application of the tabu search, can be described as follows:

1. Generation of an initial solution
2a. Route improving procedures from the initial solution consisting of a sequence of operations towards a local minimum
2b. Route procedures (improving ones, whenever it is possible) consisting of a sequence of operations from the local minimum found in 2a.

In order to apply tabu search, it is necessary to define how moves in a tabu list will be represented. Glover [8,9] mention that for strategic and computational reasons, it is usually convenient to represent moves only by a partial range of its descriptive elements or attributes. This means that each element of the tabu list will not be associated to one move but to sets of moves that share the same attributes. From this perspective, the first step to define a tabu list for a hill climbing procedure like the one presented, it is to determine the attributes which describe the movement implicit in such operations.

An early application of tabu search to the Travelling Salesman Problem provides a good example of such description [8,9]. In this work it is applied an arc interchange method - the 2-OPT heuristic. This heuristic is characterized by the substitution of two non-adjacent edges that don't belong to the current solution for other two that do belong. It is possible to define the move implicit in such operations by identifying the added edges and the deleted edges. For this reason, it can be assumed two attributes: the added edges and the deleted edges.

In node interchange operations, on the other hand, such characterization may require more complex functions to provide a reasonable number of information about the move. Generally, this information consists of the type of operation (insertion or exchange), the nodes involved (including neighbour nodes in order to specify the position in the route), the routes involved, etc. From a computational perspective, such function requires a larger amount of memory and time processing.

Nevertheless, it is possible to reduce this node interchange procedure to an arc interchange procedure. For example, in the figure below, the exchange between node B and E can be seen as the deletion of edges (A,B), (B,C), (D,E) e (E,F) and the addition of edges (A,E), (E,C), (D,B) e (B,F).

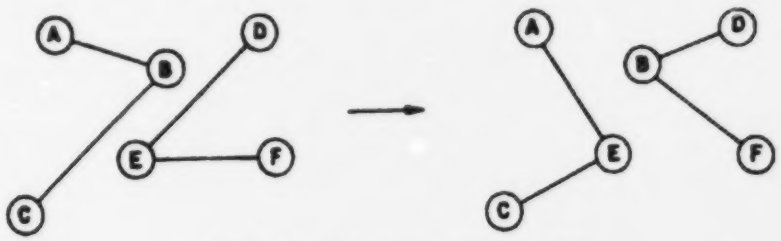

Figure 3 - Node interchange as arc interchange

Following Glover ideas, we defined the move attributes as *deleted edges* and *added edges*. Move reversals correspond to the deletion of added edges and the addition of deleted edges. We created two simple tabu lists; in the first (DEL) it is stored the identifiers of the deleted edges and in the second (AD), the identifiers of the added edges, in the last t iterations. The restrictive element works in these lists, identifying the edges characterized by a given attribute. The edges receive a prohibited tag concerning the other attribute in the following t iterations. It means that the addition (deletion) of deleted (added) edges in a given movement will be subject to restrictions during the next t iterations. An edge is considered tabu if the application of a movement implies in its inclusion in one of the tabu lists, given that it already belongs to the other list.

The movement status is defined as:

> Let E(s) be the set of deleted and added edges in a given movement s. Let $E_t(s)$ be the subset of E(s) such that its elements are tabu. A movement s is tabu if the number of element of $E_t(s)$ is greater than L, where L is an integer less or equal to the sum of the deleted and added edges.

We defined the movement status by the evaluation of the status of the edges involved in the movement. A movement is tabu if the number of tabu edges exceeds a certain threshold previously defined. By applying a range of values for L and t, it is possible to define the limits between harshness that prevents good solutions and complacency that generates cycling.

In step 2 of the algorithm's modified version we will be applying the strategy of tabu search. In step 2a the two lists AD and DEL were created to store the movements reversals. As the operations performed in this step are strictly

positive, there is no way to return to a previous state solution (the solution cost decreases monotonically). Therefore, the status of the candidate movements are not evaluated.

In step 2b, the evaluation is done at each iteration since we have crossed the first local minimum.

4 COMPUTATIONAL RESULTS AND CONCLUSIONS

In this work we applied the modified version of the route improving algorithm to three sets of routing problems. These sets are basically characterized by the existence (or absence) of time constraints, and can be stated as follows:

Set i - There is no time restrictions.

Set ii - It is defined a maximum route time and service time for each customer.

Set iii - It is defined a maximum route time, service time and time windows for each customer.

Sets i and ii are classical difficult problems whose details can be found in [2] and [3]. They consist of problems of 50, 75, 100 and 120 customers. Set iii consists of problems whose data were randomly generated. The size of problems in this set ranges from 50 to 200 customers.

The characteristics of the test problems are summarized in table 1.

Set	Problem Number	Problem Size	Vehicle Capacity	Maximum Route Time	Service Time	Time Window Range
i	1	50	160	∞	0	∞
	2	75	140	∞	0	∞
	3	100	200	∞	0	∞
	4	150	200	∞	0	∞
	5	120	200	∞	0	∞
	6	100	200	∞	0	∞
ii	7	50	160	200	10	∞
	8	75	140	160	10	∞
	9	100	200	230	10	∞
	10	150	200	200	10	∞
	11	120	200	720	50	∞
	12	100	200	1040	90	∞
iii	13	50	100	800	120	300
	14	50	100	800	120	120
	15	100	200	1200	120	300
	16	100	200	1200	120	210
	17	150	300	1400	120	300
	18	150	300	1400	120	210
	19	200	300	1400	120	400
	20	200	300	1400	120	600

Table 1 - Characteristics of test problems

4.1 DEFINITION OF L AND t VALUES

We first analyzed the algorithm behavior by varying the maximum number of tabu edges acceptable in a movement (L) and the length of tabu lists (t). We defined an initial range of L between 0 and 6, and t equals to 10, 20, 30 and 40. Applying these tests to problems 2, 3, 5 and 12, we found a typical behavior that can be represented as follows:

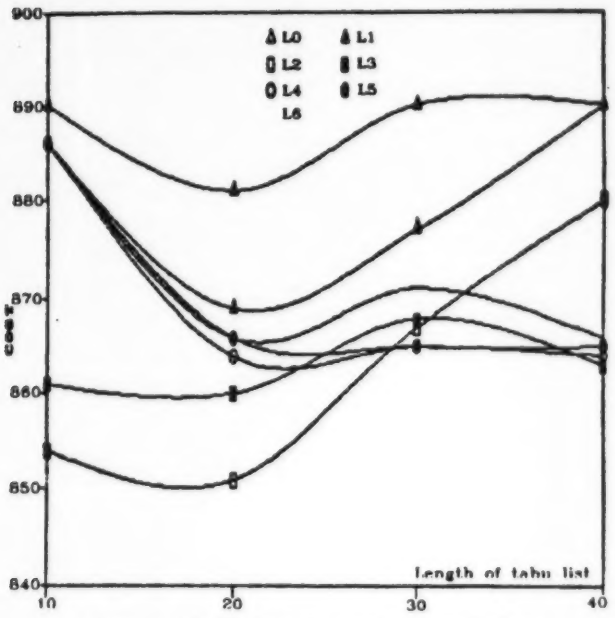

Figure 4 - Solution costs as a function of L and t values

The best solutions were found for intermediate values of L (2, 3 and 4) with list lengths between 20, 30 and 40.

For small values of L, the method does not provide good solutions, specially when we apply large list lengths t. As mentioned by Glover [8,9], greedy strategies, like the one presented, tend to focus on generating good solutions that usually display common decision elements like certain edges or groups of edges. In such restrictive situations (small L), after a first appearance in the initial stages of the search, the edges are usually prohibited to be incorporated until they regain the status of free elements. During this period, the method has no choice but incorporate edges that have not been included for not providing low cost solutions. The larger list length, the later liberation of desirable edges leading to an ineffective search process that tends to generate bad quality solutions. For small list lengths, better solutions were found but, even in this case, it seems to be necessary a large number of iterations to reach the best one.

On the other hand, given their unrestricted features, we noticed the occurrence of cycling for large values of L (5 and 6). As a result, the solutions tend to be high costly, specially when we apply small list lengths. We also found cycling for all values of L with $t = 10$.

The figure below represents the search process for problem #2, with $L = 2$ and $t = 20$.

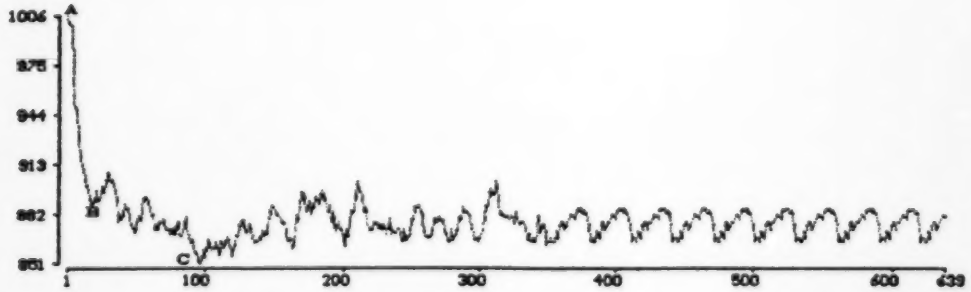

Figure 5 - A typical search process

Point A indicates the initial solution cost (Step 1). Point B indicates the first local minimum achieved (Step 2a) and point C the iteration where the tabu phase found the best solution (step 2b). It should be noted that from iteration 380 on, the search cycles.

These results suggested a range of L between 2 and 4 and t equals to 20, 30 and 40 to be applied to the other problems. We did not find any relation between values of these two parameters and problems characteristics such as size or existence of time restrictions.

4.2 ALGORITHM PERFORMANCE

The table below displays the performance achieved by applying our tabu search implementation to the proposed problems.

Defining:

Z_s = cost of the starting solution
Z_f = solution cost of the first local minimum
Z_t = cost of the best solution found during tabu phase

Problem Number	$\frac{(Z_s - Z_t)100\%}{Z_s}$	$\frac{(Z_f - Z_t)100\%}{Z_f}$	CPU time* (seconds)
1	12.3	8.6	973
2	15.5	4.2	155
3	18.8	4.3	1796
4	16.7	4.6	7351
5	5.4	4.8	1776
6	8.6	6.8	3527
7	10.0	5.5	1557
8	2.8	2.5	2772
9	12.5	7.5	7245
10	7.2	2.9	19376
11	3.3	0.5	3501
12	5.9	1.9	6166
13	46,2	11.5	367
14	48,8	16.2	289
15	57,9	10.6	1810
16	57,1	18.8	1844
17	41,8	20.4	2357
18	56,7	19.6	3191
19	61,3	17.3	8323
20	59,5	7.7	9907

* Problems 1 – 12: microcomputer MC 68020
Problems 13 – 20: workstation SUN 3/60M

Table 2 - Performance of the tabu search implementation

It should be noted that our algorithm embodies the elements of the simple tabu search, as defined by Glover [8,9]. We did not incorporated aspiration level mechanisms nor medium or long term memory functions. In addition, the computation of node insertions and node exchanges is exhaustive and does not discard operations with small chances of being selected. Another feature of our implementation is that the chosen movement to be performed at each iteration is always the best one. These observations explain the large computational time required. We believe that the application of some simple heuristics as a way to reduce the amount of insertions and exchanges can lead to smaller processing times, with no effect on the solutions quality.

Table 3 displays a comparison of our heuristic algorithm with other seven heuristics, already published:

A. Clarke and Wright's algorithm
B. Mole and Jameson's algorithm
C. The sweep algorithm
D. A tree search algorithm [3]
E. A two-phase algorithm [3]

F. Fisher's algorithm [5]
G. PSA3 Parallel savings algorithm [6]
H. The tabu search implementation

Details of algorithms A, B, C, D and E and problems' data can be found in [2] and [3]. The tested problems are the first twelve problems of Table 1. The numbers in each box give the total mileage.

Problem Number	A	B	C	D	E	F	G	H
1	585	575	524*	534	547	524*	556	536
2	900	910	865	871	883	857	855	842*†
3	886	882	851	851	851	833*	860	851
4	1204	1259	1079	1064	1093	1014*	1089	1081
5	1079	1100	1266	1092	1066	–	1052	1049*
6	831	879	937	816*	827	816*	834	826
7	619	599	560*	560*	565	560*	577	560*
8	976	969	933	924	969	916*	939	929
9	973	999	888	885*	915	885*	918	887
10	1426	1289	1230	1217	1245	–	1210*	1227
11	1634	1590	1776	1608	1612	–	1552*	1631
12	877	883	949	878	876	–	874	866*

* best solution
† different initial solution

Table 3 - Comparison with other heuristics

Our method found the best solution for problems 2, 5, 7 and 12. These solutions improved the second best cost found up to 1.5%.

In respect to the cases where our algorithm did not provide the best solution, it still showed a fine performance. Concerning algorithms A, B, C, D, E and G, it seemed to be the most regular method in terms of solution quality. In seven problems it provides solution costs that did not exceed 2.3% in respect to the best solution found by the others. Only one problem showed a solution cost 5% superior to the best one (problem #11).

In respect to algorithm F, we found costs up to 2.6% superior to the available solution. In only one problem we found a solution cost that exceeded the best one in 6.6% (problem #4).

It should be pointed up that this implementation did not apply any of tabu search elaborations for efficiency increase nor any special structure for data processing. For this reason, we do believe that the results presented are a very strong evidence that tabu search is a powerful tool for solving these class of combinatorial problems.

REFERENCES

1. Bodin, L., Golden, B., Assad, A. & Ball, M., *Routing and Scheduling of Vehicles and Crews: The State of the Art*, **Computer and Operations Research**, Vol. 1, no. 2, pp. 69-211 (1983).

2. Christofides, N & Eilon, S., *An Algorithm for the Vehicle Dispatching Problem*, **Operational Research Quarterly**, Vol. 20, pp. 309-318 (1969).

3. Christofides, N., Mingozzi, A & Toth, P., *The Vehicle Routing Problem*, **Combinatorial Optimization**, pp. 315-338 (1979).

4. Dror, M. & Levy, L., *A Vehicle Routing Improvement Algorithm Comparison of a Greedy and a Matching Implementation for Inventory Routing*, **Computers & Operations Research**, Vol. 13, pp. 33-45 (1986).

5. Fisher, M. L. & Jaikumar, R., *A Generalized Assignment Heuristic for Vehicle Routing*, **Networks**, Vol. 11, pp. 109-124 (1981).

6. Altinkemer, K. & Gavish, B., *Parallel Savings Based Heuristics for the Delivery Problem*, **Operations Research** (forthcoming).

7. Glover, F., *Future Paths for Integer Programming and Links to Artificial Intelligence*, **Computers & Operations Research**, Vol. 13, pp. 533-549 (1986).

8. Glover, F., *Tabu Search, Part I*, **ORSA Journal on Computing**, Vol. 1, no. 3, pp. 190-206 (1989).

9. Glover, F., *Tabu Search, Part II*, **ORSA Journal on Computing**, Vol. 2, no. 1 (1989).

10. Solomon, M. M., *Algorithm for the Vehicle Routing and Scheduling Problems with Time Window Constraints*, **Operations Research**, Vol. 35, pp. 254-265 (1987).